CHAPTER

2

Artificial intelligence in medicine: Technical basis and clinical applications

Bradley J. Erickson

Abstract

There has been a revolution in the past decade as "deep learning" has begun to show high performance with robust reliability in the real world for many imaging tasks. Current deep learning technologies are being applied to medical imaging tasks with good results, which has prompted great interest in applying them broadly into clinical practice. This chapter describes the basic principles of deep learning methods and some common applications in medical imaging.

Keywords: Deep learning; convolutional neural network; fully connected network; loss function; residual block; generative adversarial network

2.1 Introduction

Artificial intelligence (AI) has a long history of development, beginning in the 1950s when Hubel and Wiesel[1] began mapping the optic cortex of cats and noting the functional anatomy of how that part of the brain worked. It was not long after that computational models (often referred to as "artificial neural networks" or ANNs) were created, in the hope of both understanding and confirming our model of brain function, but with the additional possibility of creating artificial brains that could perform useful work. About that same time, computer scientists built rule-based systems to assist in decision-making processes, including applications in medicine. One of the earliest such systems was designed to assist in the selection of antibiotics based on gram staining and patient characteristics.[2]

There has been a wide variety of technologies used to aid in the diagnosis of medical images. The earliest report[3] relied on the physician to identify features and those were conveyed to an algorithm that then suggested a differential diagnosis. Since then, other

DOI: https://doi.org/10.1016/B978-0-12-821259-2.00002-8

© 2021 Elsevier Inc. All rights reserved.

methods that automatically extracted features directly from the pixels of the images (whether directly acquired as digital data or if captured by digitizing film) have been produced, including several FDA-cleared computer-aided diagnostic (CAD) tools.

2.2 Technology used in clinical artificial intelligence tools

There has been a revolution in the past decade as "deep learning" has begun to show high performance with robust reliability in the real world for many imaging tasks. The earlier machine learning tools were often "brittle" meaning that they failed if the input data or images did not appear very similar to the training examples, and those training examples often had to be acquired under carefully constrained conditions, limiting their application in the real world.

Current deep learning technologies can be divided into several families, largely depending on the dominant technological feature. The first point to make is that deep learning gets its name because its networks have many layers. For instance, fully connected networks (FCNs) are the prototypical neural networks, where the layers consist of multiple nodes, each "connected" to the subsequent layer by a "weight." Convolutional neural networks (CNNs) begin with several layers that perform convolutions on the input, each of them often followed by pooling layers that combine features while reducing the resolution of an image. Recurrent neural networks (RNNs) have a connection from a later layer to an earlier layer in the network, accounting for the "recurrent" nature pointed to in the name, and these are often applied to situations where the input data is repetitive.

It is not possible to cover every component of modern deep learning algorithms in this chapter, and instead it will be focused on clinical applications. If the reader is more interested in these other elements, one recent review that covers them in greater depth is found at Ref. [4].

2.2.1 Elements of artificial intelligence algorithms

There are several common components that are often seen in different types of deep learning algorithms. This section describes these elements, and later parts of the chapter then describe how they are combined to address problems in specific ways.

2.2.1.1 Activation functions

Activation functions are based on the action potential seen in neurons. While action potentials in biological cells are binary (either they have "fired" or not), computational "neurons" (also referred to as "nodes") can output either binary or numeric values. They can also be more flexible about the basis for computing its output, including summing the output but other options include using the slope (rate of change) of inputs as well as more complex functions of the inputs. In addition, biological neurons have a limited firing rate, with a refractory period after firing, while computational neurons are limited only by the speed of the computer.

There are several properties of activation functions that are useful for learning systems. The properties include being:

1. Nonlinear. If the output is a linear function of the inputs, then the output can only be a linear function as well.
2. Differentiable. While not required in all cases, most learning systems require the calculation of gradients in order to update the weights as the optimal set of weights is pursued. Some special functions like rectified linear units (ReLU) are differentiable everywhere except at 0 and that can be efficiently handled programmatically.
3. Controlled range. In most cases, it is desirable to limit the range of values that are output by the activation function, else the systems often become unstable.

It is not necessary to use the same activation function for all nodes. It is common practice in current deep learning systems to use a function like ReLU for most of the layers but use a function like SoftMax at the output layer to produce values something more akin to a probability.

2.2.1.2 Fully connected layer

Fully connected layers (sometimes also called dense layers) are the prototypical layers that compose a neural network. Each layer consists of an arbitrary number of nodes (neurons), which receive weighted inputs from the prior layer, add those values up (in rare cases, other operations like rates of change or product have been described), and output a value based on that sum. The output value is determined by the activation function. This computational model was first proposed by McCulloch and Pitts in 1943[5] but the concept of a complete neural network that could learn is generally attributed to Turing.[6] Simple neural network consisting only of fully connected layers is shown in Fig. 2.1.

2.2.1.3 Dropout

A challenge with deep neural networks is that they can overfit, that is, the network learns features that are specific to the examples in the training set, rather than the general features present in the class of examples in either training or testing sets. There are several ways to address overfitting, and one that is an architectural approach is called dropout.

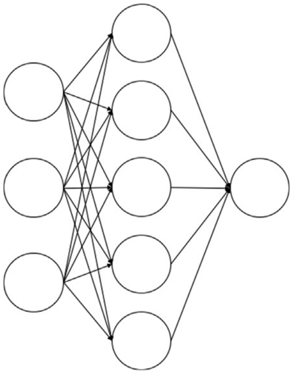


FIGURE 2.1 A simple neural network consisting of an input layer with three nodes, a hidden layer with five nodes, and an output layer with just one node.

This is a technique described by Hinton et al.[7] and involves the exclusion of randomly selected nodes with each presentation of a training sample. It is common to use a dropout ratio of 0.5, meaning that 50% of nodes are excluded. Dropout is thought to be a more efficient way to implement many networks (each set of dropped nodes is effectively a unique network) into one computational framework.

2.2.1.4 *Residual blocks*

Residual blocks[8] (also known as "residual networks" though this should be used to refer to complete networks that incorporate residual blocks) consist of typically two to three fully connected layers that have a "skip connection" from the input to the small group of layers to the output of the layers, thus forming the "block" (see Fig. 2.2). Using residual blocks almost always improves performance because it forces each layer to learn—in the case that a layer within the block does not do better than what it receives as input, the skip layer is selected. It has been shown that residual blocks smooth the gradient space. Architectures with multiple skips to multiple layers are often referred to as DenseNets.[9]

2.2.1.5 *Initialization*

Neural networks have many weights—essentially 1 for every connection between each node in a layer and the subsequent layer. Therefore a typical neural network will have many thousands to millions of weights that are to be learned during training. This raises the question: what values should the nodes have at the start? One might first consider initializing all weights to be 0 (or any other single value), but in that case the derivative for the loss function (the summed output error) will be the same for every weight in a given layer. Therefore one cannot effectively update the weights, and the network will not learn.

Random values are frequently used to initialize the weights, but it has been found that using any random value can lead to poor learning. Instead, constrained random weights seem to work best, and there are a few options that are commonly used. One simple option is to limit the range to be from −1.0 to 1.0, also referred to as random uniform and one with a normal distribution (random normal). Somewhat better performance can be

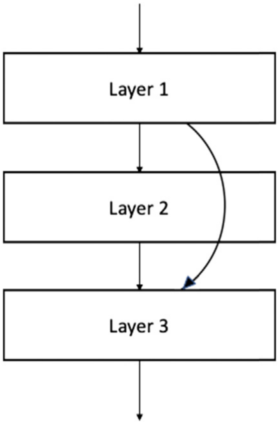


FIGURE 2.2 A residual block. The output from layer 1 is fed both to layer 2 and to layer 3. If layer 2 is not able to do better than its input, layer 3 will effectively ignore it, thus reducing the number of weights in the network, and thus reducing the chance of overfitting. Note that there can be more layers between the skip connection (e.g., a layer 2a between layer 2 and layer 3) but typically there are only one or two layers plus the output layer in the residual block.

shown with He, Xavier (also known as Glorot) initialization, or LeCun functions, all of which have normal and uniform distribution forms.

One special form of initialization is transfer learning. If there are networks that have already been trained on problems similar to your task, it is often more efficient to take the network (including trained weights) and begin by training that network. It is likely that the learned weights will be a better starting point than any of the initialization functions described previously. It is also common practice to "freeze" the weights on early layers, based on the assumption that the low-level features are common, if the task truly is similar. Transfer learning can be particularly useful when your training set is limited, because freezing layers reduces the number of weights to be learned.

2.2.1.6 Convolution and transposed convolution

A convolution is a core function that is used for extracting features from the input image or signal. Convolution consists of having a convolutional kernel (which is a small matrix that matches the dimensionality of the input function) that is moved across the input image and corresponding elements of the image and kernel are multiplied, and then each of those products is added to produce an output. For example, if the input is a 2D image, the kernel is also 2D, such as a 3×3 matrix. In this case, the output image will have the same dimensions as the input image. Another strategy is to not convolve the edge pixels, in which case the output image will be smaller than the input. Increasing the matrix size of an image (effectively magnifying it) is often needed. A simple way to do this is to duplicate each pixel in each dimension, or a next level of sophistication is to linearly interpolate the values. However, both of these strategies can lead to undesirable effects. Transposed convolution (sometimes incorrectly referred to as deconvolution or correctly referred to as upsampled convolution) is a technique to increase the matrix size of an image by using a kernel. In this case, as the kernel is passed over the input image, each element of the kernel is multiplied by the corresponding pixel in the image plus input image neighbors to produce a component the size of the window. This is repeated for each element of the kernel, and then all the components are added together, producing an upsampled image that reflects features the kernel was selected to amplify.

2.2.1.7 Inception layers

GoogLeNet contains multiple *inception modules*, in which multiple different filter sizes are applied to the input and their results concatenated. This multiscale processing allows the module to extract features at different levels of detail simultaneously. GoogLeNet also popularized the idea of not using fully connected layers at the end, but rather global average pooling, significantly reducing the number of model parameters.

2.2.2 Popular artificial intelligence software architectures

2.2.2.1 Neural networks and fully connected networks

Previous sections have described fully connected layers where each node of a layer is connected to each node of subsequent layers by a weight. A FCN will usually include many such layers, often with ReLU activation functions for most layers (often with

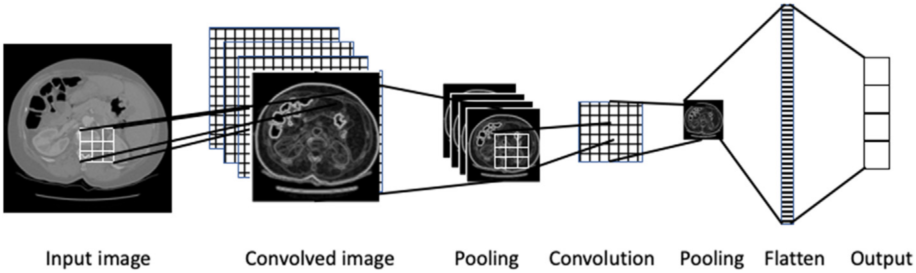


FIGURE 2.3 Simple convolutional neural network. The input image is convolved with a kernel. The convolved image is then reduced in resolution using pooling. There are usually 3–10 groups of convolution and pooling applied. The last pooled image is then "flatten" meaning that the 2D array is converted to a 1D array that can be used as input to fully connected layers until an output is predicted.

dropout), but a sigmoidal-shaped function as the output, which creates output values that are more akin to a probability, which is helpful in understanding how to convert values to decisions.

2.2.2.2 *Convolutional neural networks*

CNNs are a popular form of deep neural network that have proven quite effective for image-focused learning. A fundamental difference between a CNN and a fully connected network is that the first few layers consist of alternating convolutions and pooling (Fig. 2.3). After these initial layers the 2D image is "flattened" into a 1D array of values and that is then usually used as input to a few layers of fully connected network until the final output layer is reached.

Convolutional neural networks work well for many image (and also some 1D problems) where the location of an object in the input is not known, and where the thing of interest consists of several parts that must be combined together to be recognized, and where precise size and scale are not known. During the learning process, both the values in the convolutional kernel (also referred to as a "filter") and the weights of the fully connected layers are updated/learned. In general, the convolution kernels are learning the features in the images that are important while the fully connected layers learn the best weights and combinations of those features. It should also be noted that there are many convolutional kernels at each resolution, so many different features can be found at each level of resolution. While radiological images are usually gray scale, photographic images are red/green/blue, and CNNs will also have separate kernels for each color channel. As a result, even if we specify just a 5×5 kernel in a certain layer, there will be many more values because of the number of color channels and the number of filters per layer. For example, if a 5×5 kernel is used with 3 color channels and 16 filters, we will actually have $5 \times 5 \times 3 \times 16 = 1200$ values that can be learned for that 1 layer.

After each convolutional layer, there is often a pooling layer, which serves to reduce the resolution of the image. Commonly the maximum pool or MaxPool function is used. It is a simple function that takes a small region of an image (such as 2×2) and outputs the maximum value in that region. This effectively reduces the resolution of the image, and taking the maximum (though some also use the mean value) is thought to be a good function because it rewards regions where the kernel has found a feature it matches.

2.2.2.3 *U-Nets and V-Nets*

Segmentation is the assignment of labels to pixels of an image. For instance, if the image is a CT scan of the abdomen, a segmentation algorithm might identify the liver or tumors within the liver. Segmentation is a critical step in many image analysis tasks and has been the focus of decades of research. Early methods focused on finding edges or regions of similar intensity but these were often very sensitive to how the image was acquired and also required that the item being segmented have a consistent appearance—something often not true when pathology is present.

A novel deep learning architecture designed for segmentation is the U-Net.[10] It gets its name from the architecture (Fig. 2.4) wherein the first steps of the algorithm successively reduce the resolution of an image while extracting key high-level properties of the object being segmented. At the lowest resolution (the bottom of the "U") the key features of the structure should be captured. The right-hand side of the U consists of restoring the resolution of the image while retaining focus on the structure of interest. There are "skip connections" where the corresponding resolution of the input image on the left side of the "U" is accessed by the right-hand side to assist in localizing the margins of the object of interest.

A typical U-Net will use many of the components described previously, including convolutional elements as well as pooling elements that reduce resolution. It then uses transposed convolutions to increase the resolution on the right side but uses "skip connections" from the convolutions.

The original description of the U-net was for 2D biological images, but this has now been successfully applied to many other 2D images. It has also been extended to 3D, which is referred to as a "V-Net."[11]

2.2.2.4 *DenseNets*

ANNs were inspired by what we understood to exist in the brain. In this arrangement, layers of nodes (neurons) are sequentially connected to subsequent layers of nodes. This

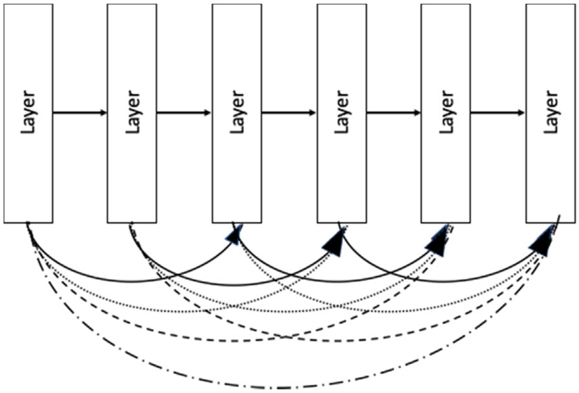


FIGURE 2.4 A DenseNet. This is similar to a group of residual blocks, but with the distinction that each layer has skip connections to all subsequent layers, while the prototypical residual block just skips one or two layers.

leads to a clean and understandable computational model that has been applied to many problems. A challenge with this architecture is the *vanishing gradient problem*, that is, when an error is computed at the output of a network, it can be hard to decide how the error should be used to update a given individual layer: if there are 20 layers, should the error be evenly distributed across each layer? Typically the error is apportioned according to the weights in a layer so that layers with small weights do not get a large change while layers with large weights are effectively "ignored." Each of the neural network's weights receives an update proportional to the partial derivative of the error function with respect to the current weight in each iteration of training. The problem is that in some cases, the gradient will be vanishingly small, effectively preventing the weight from changing its value.

The residual block has already been described and addresses the challenge of assuring that each layer contributes positively to the performance of the network, by forcing it to compete with the identity function. The next logical extension is to connect a given layer to all of the subsequent layers, thus forcing each layer as well as the sum of all layers to do better than identity. Network where a layer is connected to all (some might say "many") subsequent layers is referred to as a DenseNet[12] (see Fig. 2.4). The original description had convolutional layers at the start, but some have also described a DenseNet without convolutions.

Because the skip connections force more efficient learning, it is common to reduce the number of nodes in each layer (making the layer "narrower"). Besides better parameter efficiency, another advantage of DenseNets is their improved flow of information and gradients throughout the network, which makes them easier to train. Each layer can directly access the gradients of the loss function, which helps training of deeper network architectures.

2.2.2.5 Generative adversarial networks

Generative adversarial networks (GANs) were first described by Ian Goodfellow[13] and represent a distinct departure from the typical learning paradigm. While most deep learning architectures focus on making predictions about existing images based on patterns learned from other images, GANs attempt to learn about existing images in order to create new images or pieces of images. Since that original design, many useful variants have been described.

The basic GAN (Fig. 2.5) consists of a generator that typically has some source of variance such as a noise generator. (This is more frequently described as a "latent space" and other variants of GANs control this in order to achieve desired outcomes.) The generator is tasked with learning to generate images that simulate the collection of real images. The second main component is a discriminator that is essentially a classifier that tries to determine whether an image presented to it is real or fake. These two elements compete against each other (hence, the term "adversarial") with the hope that the generator will learn to create very realistic images.

2.2.2.6 Hybrid generative adversarial network designs

Since the original description of the GAN, there have been a number of variations that have proven to be useful for medical imaging. Several variations have been shown to be effective for creating images of one modality or contrast type based on a different input image. Some examples include converting MRI to CT for purposes of attenuation correction in treatment planning or PET imaging. There are also examples converting different

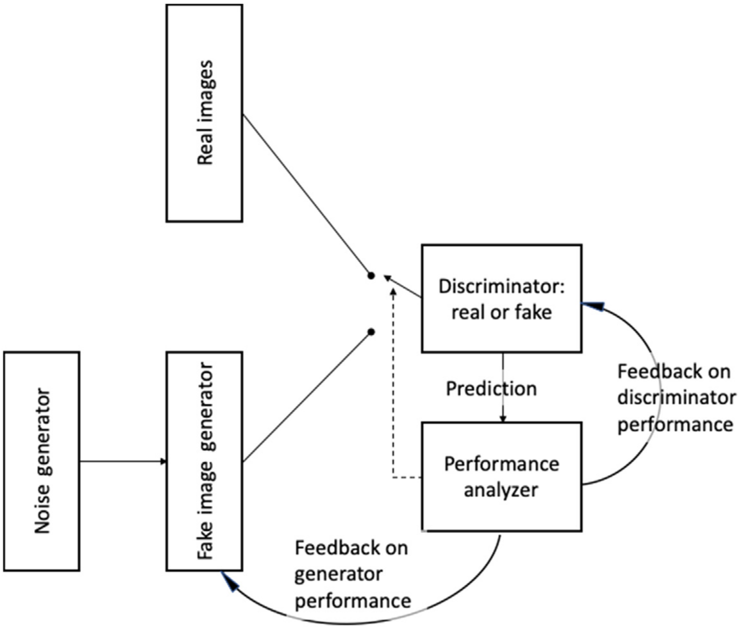

Real images
Discriminator: real or fake
Prediction
Feedback on discriminator performance
Noise generator
Fake image generator
Performance analyzer
Feedback on generator performance

FIGURE 2.5 Basic GAN architecture. There is a collection of real images, which the GAN should learn to simulate. There is also a fake image generator that uses noise and feedback about its performance to generate images that should be like the real images. The discriminator tries to classify an input image as real of fake. A performance analyzer knows whether a real or fake image was presented to the discriminator and gives feedback to the discriminator about its performance. It also gives feedback to the generator about how well it is fooling the discriminator. *GAN*, Generative adversarial network.

MR contrast types from one to another. It is likely that these will become more commonly used in clinical practice.

It has also been demonstrated that augmenting data sets using GANs is more effective than simple geometric variants[14,15] and this will likely put GANs in a more prominent role in medical imaging. Closely aligned with this is the ability to create specific forms of images such as examples that are normal and others that have a known disease but which are not any specific patient image. In addition to more effective training of the AI algorithm, this can avoid challenges surrounding patient privacy. It is likely that this use of GAN technology will become more widespread.

2.3 Clinical applications

2.3.1 Applications of regression

The output of an AI algorithm can take many forms, but perhaps the simplest is to perform regression, which means that the output is a floating-point value. In this case the output layer may be as simple as an addition function of all the outputs of the nodes in the prior layer.

2.3.1.1 Bone age

One of the earliest applications of AI in medical imaging was the prediction of own age based on a hand radiograph. This particular task is used in the case where children seem to be either more advanced or delayed compared to their chronological age and the state of maturation of the bones of the hand turns off and reflects the metabolic state of the patient. In the past, there have been radiographic atlas that humans would then use to compare a patient hand radiograph versus the large collection of normal hand radiographs. This was also a good task for AI because creating the gold standard was easier, in that the reporting format for pediatric hand radiographs was very stereotypical. The report typically would consist of the estimated bone age in years and months and might also include the patient's chronological age so that could be deduced from the medical record.

There are now several examples of bone age estimation by AI algorithms, in part promoted by the RSNA bone age challenge that provided a large curated data set of hand radiographs along with the bone age reported by a human expert.

2.3.1.2 Brain age

Just as hand radiographs can be used to estimate the metabolic age of a child, there are now reports using MRI imaging to estimate the brain age of subjects.[16,17] There are many large public databases of high-resolution anatomic brain images along with patient demographics (including patient age), which have been used to create algorithms that predict the age of the patient. If this can be reliably performed, looking at the tires may provide insight into patients with specific diseases. One recent report also shows that this can be helpful in identifying molecular markers.[16] It is also likely that just as one may be able to predict patient age from MRI of the brain, tools to predict patient age (and variation from the patient chronologic age) could be applied to other images such as chest CT or abdominal CT, which might in turn identify important biomarkers of disease.

2.3.2 Applications of segmentation

As noted previously, segmentation is the assignment semantic labels to pixels of an image. This is an important step for measuring size and other properties of a structure from an image. It is also often a first step before other tasks such as regression or classification are applied to an image. Because of this, segmentation has received much attention. While it is desirable to have a segmentation algorithm be as general as possible, it is now clear that the best results are obtained when information both about the type of image and also properties of the structure being segmented are known.

It is not possible to review all deep learning−based literature on segmentation of medical images. It is worth noting that the U-Net[10] was first described for biological images, and is now applied not only in medicine, but is widely used outside of medicine and biology. A query of PubMed literature, including the term "U-Net" or (deep learning and segmentation) returns 184 publications during 2019, 352 during the prior 5 years, and only 4 additional publications prior to 5 years ago. This shows the rapid adoption and acceleration of this technology to the field.

The U-Net has already been described, but it is worth paying attention to the error metrics used in evaluating segmentation methods. Probably the most commonly used error metric is the Dice similarity coefficient (DSC)—it is popular enough that it is built into many popular deep learning frameworks. The DSC is 1.0 when the predicted segmentation perfectly matches the ground truth, and is 0 when none of the ground truth pixels are correctly identified. This function can work well for many applications, but a few caveats should be noted:

1. In some medical applications, it may be more important to assure that one is never more than a certain distance from the correct outer margin of a structure, and in that case the Hausdorf distance is a better metric. For very large structures (e.g., the liver), one can get near-perfect DSCs but one could still have a few pixels that are a great distance from the true margin. Conversely, for objects with a high surface area:volume ratio, an error in segmenting just a few pixels can result in a very poor score, even if that error is just pixels on the edge of the structure (errors that might result from partial volume effect).
2. In cases where multiple objects are being simultaneously segmented, it can be challenging to get the right weighting of the DSCs of each object. If the objects include both large round objects like the liver that easily result in high DSCs and small structures like adrenals that are challenging to obtain high DSCs, using a performance metric that averages all the DSCs together could result in very poor liver segmentation in order to get reasonable adrenal segmentation.

Therefore it is important to understand the medical drivers of the segmentation task in order to determine the optimal error metric and acceptable performance.

The list of applications of deep learning for medical image segmentation is very long and rapidly growing. There have been several international challenges for image segmentation such as those organized by the MICCAI, ISBI, and SPIE (see https://grand-challenge.org/challenges/), which have substantially accelerated the number of papers published for specific segmentation tasks.

A major driver for publications and success in image segmentation is the availability of large data sets required for training a deep learning system. As such, it should not be surprising that frequently imaged organs would be among the most popular: brain, lung, heart, liver, and kidneys. In addition to these organs, there are also good results being shown for segmentation of cancer of some of these organs, including tumors of the brain, lung, and liver, which approach human-level performance.

2.3.3 Applications of classification

Classification models attempt to predict a label or class for a given example. It may predict the class using a continuous value that is the probability of a given example belonging to a class. If the classification problem is binary ("yes" or "no"), then one can simply set a threshold of 0.5 to make the decision. If there are multiple classes, the probabilities can be interpreted as the likelihood or confidence of a given example belonging to each class. A predicted probability can be converted into a class value by selecting the class label that has the highest probability.

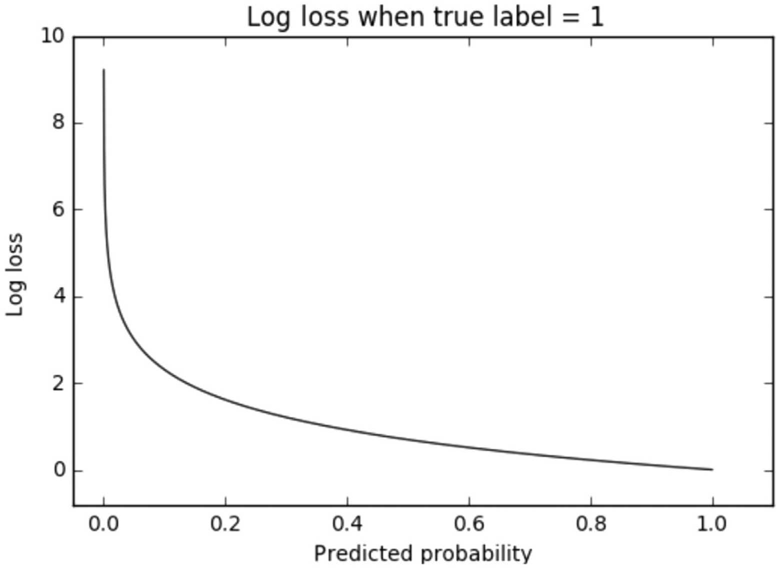
In binary classification, where the number of classes equals 2, cross-entropy can be calculated as:

$$-(y\log(p)+(1-y)\log(1-p))$$

If M > 2 (i.e., multiclass classification), we calculate a separate loss for each class label per observation and sum the result.

$$-\sum_{c=1}^{M} yo,\ c\log(po, c)$$

where M is the number of classes (normal, tumor, necrosis); log is the natural log; y is the binary indicator (0 or 1) if class label c is the correct classification for observation o; p is the predicted probability that observation o is of class c.



2.3.3.1 *Detection of disease*

There are now hundreds of reports in the peer-reviewed medical literature on the use of deep learning methods to detect disease. The vast majority of these are for the detection of a specific disease using a specific imaging modality. Peak performance of most of these approaches that of human experts and in some cases surpasses it. This has sometimes lead pundits to suggest that computers will replace physicians where at least some diagnostic tests. It should be noted that while a computer can be useful for the detection of some specific diseases, there are few reports on the detection of most or all diseases in a given type of image and that represents a fundamental challenge to the replacement of humans by computers. One recent publication demonstrates the application of AI to detect a broad range of findings in imaging.[18] It shows that for most classes of findings, the AI algorithm was able to perform similar to that of human experts. That group and others, however, note that the most productive model is where the AI supplements the human expert rather than attempt to compete separately. An example of this application of AI to medicine is seen, for example, in Ref. [19].

2.3.3.2 *Diagnosis of disease class*

Just as an AI tool can be trained to determine whether or not a specific finding is present, it can also be trained to differentiate between two classes of findings. One common challenge in the practice of medicine is to determine whether or not a patient is responding to a specific therapy. In this case the challenge is not to diagnose the disease but rather to assess the response. Most recent publications on the use of AI in medical imaging are binary classifiers—a specific disease or finding is present or not. These are usually narrow clinical questions, and the wide variety of possible imaging findings limits the usefulness of these tools.

One good example showing broad coverage of AI findings is for detection and classification of abnormalities on chest radiographs.[18] In this study of over 100,000 chest radiographs, findings were grouped into 14 categories covering most of the findings that might be observed. They then built a multiclass classifier to identify the findings. The classifier performed at a level similar to radiologists, with the exception of emphysema. Even with this broad coverage, it should be noted that not all important findings were included, for example, detection of gas under the diaphragm and bone fractures is important if rare.

2.3.3.3 *Prediction of molecular markers*

One of the most exciting applications of deep learning to medical imaging is the ability to identify important molecular markers from routine imaging. Radiology in particular has not participated in the genomics revolution to any significant degree, but deep learning could change that. There are now several reports on the ability of deep learning algorithms to detect features in routine radiological images that predict important molecular markers with high accuracy. Brain glioma markers are probably the most advanced with greater than 90% accuracy for predicting such markers as isocitrate dehydrogenase, 1p19q chromosomal deletion, and MGMT methylation status.[20,21] Other reports show good performance for prediction of some lung cancer molecular markers[22,23] as well as molecular markers associated with various forms of dementia.[24−26] It is likely that many more molecular markers will continue to be developed over the next few years for many diseases.

2.3.3.4 *Prediction of outcome and survival*

While the prediction of molecular markers from radiographic imaging is revolutionary, in some respects, it may not be the best ultimate target. It is known that complex gene−gene interactions as well as host factors can alter the expression of a single molecular marker. Since imaging "sees" the phenotype of the tumor, it is possible that predicting the responsiveness to a therapy or the likely survival of a patient may be more important. Indeed, several groups have now also shown that radiomics can make reasonably accurate predictions of clinical course, which may be independent of the markers. This has been shown for diseases such as head and neck cancer,[27] lung cancer,[28] and polycystic kidney disease.[29]

2.3.4 Deep learning for improved image reconstruction

The design of imaging devices has always included mathematical formulas that produce images based on an understanding of the physics of the device. There are now many reports on the use of deep learning methods to create the images both from the raw detected signal, as well as to improve the quality of the image. A common element of these algorithms focuses on the creation of "full quality" images from reduced signal. The reduced signal may be less dose in the case of X-ray-based modalities such as CT, or less RF signal such as in MRI where less time is needed to collect the image data. In most cases the learning algorithm is given the reduced signal either in the form detected, or possibly with some component of traditional reconstruction performed (e.g., Fourier transform for MRI) and the algorithm then attempts to create the "full quality" image from that limited signal input.

In the case of CT, one can train a CNN to work as a filter to reduce the noise in low-dose images such that they look similar to full-dose images,[30] and these methods perform well or better than traditional filtering methods.[31]

CNN- and RNN-based image reconstruction methods are rapidly increasing, including methods that directly reconstruct MRI from limited k-space data.[32,33]

2.4 Future directions

2.4.1 Understanding what artificial intelligence "sees"

It is a common misperception in the medical community that AI tools are a "black box" and that one cannot understand how they work. For traditional machine learning methods, this is particularly untrue, as humans select the features, and the relative weighting of those features is discernible. For deep learning, determining the features is more challenging, but not impossible. In the case of CNNs, one can both directly observe the values used in the kernels of the convolutions and also see the activations of the network for a particular input, and there are publicly available toolkits that can allow any user to visualize these activations.[34] It is becoming rather common for publications to include saliency maps, which focus on the impact of input pixels on making a decision, as well as activation maps that reflect the gradients in the final layers of a network, which also reflect important parts of an image. A description of these is provided in Philbrick et al.[35] Because many applications of deep learning require confidence in the decision basis, improvements in making the "black boxes" more transparent will continue.

2.4.2 Workflow

A challenge to the implementation of AI tools into clinical practice is assuring that all required information is efficiently provided to the tool. Most AI tools today are simple and require only one image. However, AI always does better with more information, and it is almost certain that AI tools of the future will provide better decisions by taking advantage of richer input data. This may include more images—ones with different contrast properties, or from different time points. It may also include more nonimage data, such as patient demographics, known diagnoses, prior and present therapies, and the times of these therapies. All of these are known to improve human performance for most medical tasks, and it is certain to improve AI performance as well. This demand for richer input data will require more sophisticated AI architectures, improved data curation methods, and better clinical implementation environments. At present, there is limited literature on support for more complex workflows in medical imaging,[36] but AI is likely to drive broader adoption and development of this type of technology.

2.5 Conclusion

AI, and deep learning methods in particular, has made significant advances that have resulted in dramatic advances for medical imaging in recent years, and the rate of adoption into clinical practice is likely to accelerate. Current challenges to broad adoption include large, diverse, and well-curated data sets required for training these systems. Regulatory, legal, workflow, and financial constraints are also impediments to adoption in many jurisdictions. However, these tools have demonstrated the ability to significantly improve the efficiency of medical care, to extract information that humans cannot perceive, and to do this in an objective fashion. This will insure that the demand for these tools will increase, as our understanding of optimal training and implementation models improves.

References

1. Hubel DH, Wiesel TN. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J Physiol* 1962;**160**:106–54.
2. Yu VL, Fagan LM, Wraith SM, et al. Antimicrobial selection by a computer. A blinded evaluation by infectious diseases experts. *JAMA* 1979;**242**:1279–82.
3. Lodwick GS, Haun CL, Smith WE, Keller RF, Robertson ED. Computer diagnosis of primary bone tumors. *Radiology* 1963;273–5. Available from: https://doi.org/10.1148/80.2.273.
4. Lundervold AS, Lundervold A. An overview of deep learning in medical imaging focusing on MRI. *Z Med Phys* 2019;**29**:102–27.
5. McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 1943;115–33. Available from: https://doi.org/10.1007/bf02478259.
6. Turing AM. Intelligent machinery. NPL. Mathematics Division; 1948. <https://weightagnostic.github.io/papers/turing1948.pdf>.
7. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. *Improving neural networks by preventing co-adaptation of feature detectors*. arXiv [cs.NE]. 2012. <http://arxiv.org/abs/1207.0580>.
8. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Bajcsy, editor. *Proceedings of the IEEE Conference on Computer Visions and Pattern Recognition*. Los Alamitos, CA: Conference Publishing Services; 2016. p. 770–8.
9. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017. 4700–8.
10. Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF, editors. *Medical image computing and computer-assisted intervention – MICCAI 2015*. Springer International Publishing; 2015. p. 234–41.
11. Milletari F, Navab N, Ahmadi S-A. *V-Net: fully convolutional neural networks for volumetric medical image segmentation*. arXiv [cs.CV]. 2016. <http://arxiv.org/abs/1606.04797>.
12. Huang G, Liu Z, van der Maaten L, Weinberger KQ. *Densely connected convolutional networks*. arXiv [cs.CV]. 2016. <http://arxiv.org/abs/1608.06993>.
13. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, editors. *Advances in neural information processing systems 27*. Curran Associates, Inc.; 2014. p. 2672–80.
14. Han C, Murao K, Satoh S, Nakayama H. *Learning more with less: GAN-based medical image augmentation*. arXiv [cs.CV]. 2019. <http://arxiv.org/abs/1904.00838>.

15. Frid-Adar M, Klang E, Amitai M, Goldberger J, Greenspan H. Synthetic data augmentation using GAN for improved liver lesion classification. In: *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. 2018. 289−93.
16. Jonsson BA, Bjornsdottir G, Thorgeirsson TE, et al. *Deep learning based brain age prediction uncovers associated sequence variants*, *Nat Commun* 2019. Available from: https://doi.org/10.1101/595801.
17. Cole JH, Poudel RPK, Tsagkrasoulis D, et al. Predicting brain age with deep learning from raw imaging data results in a reliable and heritable biomarker. *Neuroimage* 2017;**163**:115−24.
18. Rajpurkar P, Irvin J, Ball RL, et al. Deep learning for chest radiograph diagnosis: a retrospective comparison of the CheXNeXt algorithm to practicing radiologists. *PLoS Med* 2018;**15** e1002686.
19. Patel BN, Rosenberg L, Willcox G, et al. Human-machine partnership with artificial intelligence for chest radiograph diagnosis. *NPJ Digit Med* 2019;**2**:111.
20. Korfiatis P, Zhou Z, Liang J, Erickson BJ. Fully automated IDH mutation prediction in MRI utilizing deep learning. In: Erickson BJ, Siegel EL, editors. *Proceedings of the second conference on machine intelligence in medical imaging*. 2017. 23.
21. Chang P, Grinband J, Weinberg BD, et al. Deep-learning convolutional neural networks accurately classify genetic mutations in gliomas. *AJNR Am J Neuroradiol* 2018;**39**:1201−7.
22. Digumarthy S.R, Padole AM, Lo Gullo R, Sequist LV, Kalra MK. Can CT radiomic analysis in NSCLC predict histology and EGFR mutation status? *Medicine*. 2019;98:e13963. Available from: https://doi.org/10.1097/md.0000000000013963
23. Rizzo S, Petrella F, Buscarino V, et al. CT radiogenomic characterization of EGFR, K-RAS, and ALK mutations in non-small cell lung cancer. *Eur Radiol* 2016;**26**:32−42.
24. Ullah HMT, Tarek Ullah HM, Onik Z, Islam R, Nandi D. Alzheimer's disease and dementia detection from 3d brain MRI data using deep convolutional neural networks. In: *2018 third international conference for convergence in technology (I2CT)*. 2018. Available from: https://doi.org/10.1109/i2ct.2018.8529808.
25. Islam J, Zhang Y. Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks. *Brain Inf* 2018;**5**:2.
26. Wang Y, Tu D, Du J, et al. Classification of subcortical vascular cognitive impairment using single MRI sequence and deep learning convolutional neural networks. *Front Neurosci* 2019;**13**:627.
27. Diamant A, Chatterjee A, Vallières M, Shenouda G, Seuntjens J. Deep learning in head & neck cancer outcome prediction. *Sci Rep* 2019;**9**:2764.
28. Hawkins SH, Korecki JN, Balagurunathan Y, et al. Predicting outcomes of nonsmall cell lung cancer using CT image features. *IEEE Access* 2014;**2**:1418−26.
29. Kline TL, Korfiatis P, Edwards ME, et al. Image texture features predict renal function decline in patients with autosomal dominant polycystic kidney disease. *Kidney Int* 2017. Available from: https://doi.org/10.1016/j.kint.2017.03.026.
30. Kang E, Chang W, Yoo J, Ye JC. Deep convolutional framelet denoising for low-dose CT via wavelet residual network. *IEEE Trans Med Imaging* 2018;**37**:1358−69.
31. Wu D, Kim K, El Fakhri G, Li Q. *A cascaded convolutional neural network for X-ray low-dose CT image denoising*. arXiv [cs.CV]. 2017. <http://arxiv.org/abs/1705.04267>.
32. Yang Y, Sun J, Li H, Xu Z. Deep ADMM-Net for compressive sensing MRI. In: Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R, editors. *Advances in neural information processing systems 29*. Curran Associates, Inc.; 2016. p. 10−18.
33. Schlemper J, Caballero J, Hajnal JV, Price A, Rueckert D. *A deep cascade of convolutional neural networks for MR image reconstruction*. arXiv [cs.CV]. 2017. <http://arxiv.org/abs/1703.00555>.
34. J Yosinski. <http://yosinski.com/deepvis> [accessed 29.12.19].
35. Philbrick KA, Yoshida K, Inoue D, et al. What does deep learning see? Insights from a classifier trained to predict contrast enhancement phase from CT images. *AJR Am J Roentgenol* 2018;**211**:1184−93.
36. Erickson BJ, Langer SG, Blezek DJ, Ryan WJ, French TL. DEWEY: the DICOM-enabled workflow engine system. *J Digit Imaging* 2014;**27**:309−13.